

# **Rozšíření modulu pro vzdálenou správu počítačových účtů v síti**

## **Advancement for PC account network managing module**

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Ostravě 16. dubna 2009

.....

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 16. dubna 2009

.....

Rád bych na tomto místě poděkovala všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Poděkování patří především panu Petrovi Kopeckému, mému vedoucímu diplomové práce Ing.Dušanu Lampartovi a vývojovému týmu firmy Linux-Box.cz, s.r.o, za odborné vedení a spoustu cenných rad.

## **Abstrakt**

Smyslem této práce je implementace a popis rozšíření stávajícího modulu PC Account. Práce má popisovat omezující vlastnosti původní verze a jejich řešení. Mezi řešení patří zejména využití relační databáze pro uložení všech informací pro konfiguraci síťových prvků v DHCP, DNS a NT doméně, zavedení profilů pro ukládání do konfigurací a synchronizaci databáze s konfiguračními soubory.

**Klíčová slova:** Webmin modul, DHCP, DNS, NT doména, PC account

## **Abstract**

The purpose of this work is to implement and describe advancement for existing PC Account module. Work should describe limiting feature of existing version of module and solutions of this features. Solutions includes using relation database for saving all kind of information for configuration of network elements like DHCP, DNS and NT domain, further using profiles for saving information into configuration files and database synchronization with configuration files.

**Keywords:** Webmin module, DHCP, DNS, NT domain, PC account

## **Seznam použitých zkratk a symbolů**

ARP	– Address Resolution Protocol
CGI	– Common Gateway Interface
CPAN	– Comprehensive Perl Archive Network
DHCP	– Dynamic Host Configuration Protocol
DNS	– Domain Name System
GUI	– Graphical User Interface
ICMP	– Internet Control Message Protocol
IP	– Internet protocol
MAC	– Media Access Control
PING	– Packet InterNet Groper
SP2	– Service Pack 2, Windows XP patch
SSH	– Secure Shell

## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
1.1	Cíle práce . . . . .	5
<b>2</b>	<b>Omezení původní verze</b>	<b>6</b>
2.1	Nemožnost diferenciací síťových prvků . . . . .	6
2.2	Nepodpora více podsítí . . . . .	6
2.3	Dlouhá odezva při zobrazení dostupnosti síťových prvků pomocí aplikace Ping . . . . .	6
<b>3</b>	<b>Použité technologie</b>	<b>8</b>
3.1	Webmin . . . . .	8
3.2	SQLite . . . . .	8
3.3	Perl . . . . .	8
3.4	CPAN . . . . .	9
3.5	Šablonovací systém HTML::TEMPLATE::Compiled . . . . .	9
3.6	DHCP . . . . .	9
3.7	DNS . . . . .	9
3.8	Záznam v NT doméně . . . . .	10
3.9	Ping . . . . .	10
3.10	Arping . . . . .	11
<b>4</b>	<b>Zacházení s konfiguračními soubory</b>	<b>12</b>
4.1	Zacházení s DHCP konfiguračními soubory . . . . .	12
4.2	Zacházení s DNS konfiguračními soubory . . . . .	13
4.3	Zacházení s passwd . . . . .	14
<b>5</b>	<b>Zavedení centrální relační databáze</b>	<b>15</b>
5.1	Uložení informací o síťových prvcích /nodes/ . . . . .	15
5.2	Zavedení profilů . . . . .	15
5.3	Doplňující informace o prvcích . . . . .	15
5.4	Zavedení skupin . . . . .	17
5.5	Detekce chybějících záznamů . . . . .	18
5.6	Synchronizace konfiguračních souborů s databází . . . . .	18
<b>6</b>	<b>Podpora podsítí</b>	<b>20</b>
6.1	Informace o podsítích . . . . .	20
6.2	Ukládání prvku do určité podsítě . . . . .	20
6.3	Přesun prvku do jiné podsítě . . . . .	20
<b>7</b>	<b>Paralelní zjišťování dostupnosti síťových prvků</b>	<b>21</b>
7.1	Samostatná aplikace Pingu mimo proces Webminu . . . . .	21
7.2	Důvody využití příkazu Arping . . . . .	21

---

<b>8</b>	<b>Webová aplikace</b>	<b>22</b>
8.1	Seznam prvků . . . . .	22
8.2	Manipulace se záznamy . . . . .	23
8.3	Správa profilů . . . . .	26
<b>9</b>	<b>Závěr</b>	<b>27</b>
9.1	Vývoj do budoucna . . . . .	27
<b>10</b>	<b>Reference</b>	<b>28</b>
	<b>Přílohy</b>	<b>28</b>
<b>A</b>	<b>Ostatní funkce pro zacházení s konfiguračními soubory</b>	<b>29</b>

## Seznam obrázků

1	Databázové schéma . . . . .	16
2	Ukázka GUI pro manipulaci se záznamy . . . . .	24
3	Ukázka GUI pro manipulaci s profily . . . . .	25



## Seznam výpisů zdrojového kódu

1	Ukázka konfiguračního souboru DNS z <code>namedb</code> . . . . .	10
2	Ukázka souboru zapůjčených adres DHCP . . . . .	12
3	Přidání nového prvku do DNS . . . . .	13
4	Editace prvku v DNS . . . . .	13
5	Výmaz prvku z DNS . . . . .	13
6	Ukázka zacházení se záznamy v NT doméně . . . . .	14
7	Ukázka záznamů v <i>passwd</i> . . . . .	14
8	Ukázka třídění podle dvou atributů . . . . .	23

# 1 Úvod

Modul PC account je praktický nástroj, který dává uživateli funkcionalitu pro kompletní správu účtů. V kombinaci s jinými moduly se nasazuje na servery zákazníků. Tento modul začal být vyvíjen roku 2006, kdy získal většinu své funkcionality. Postupem času se projevily jeho slabé vlastnosti a začal být spíše neoblíben.

## 1.1 Cíle práce

Cílem této práce je navrhnout a implementovat rozšíření funkcionality stávajícího modulu PC account, tak aby se choval jako komplexní nástroj pro konfiguraci záznamů v DHCP, DNS a NT doméně.

Toto rozšíření se týká přesunu všech informací o záznamech do relační databáze a zavedení profilů, podle kterých se ukládají informace do jednotlivých konfiguračních souborů. Tato problematika je popsána v sekci 5.

Dalším rozšířením je navrhnout a implementovat podporu více podsítí, což je popsáno v kapitole 6 a urychlení procesu zjišťování dostupnosti síťových prvků, viz sekce 7.

Dalším cílem práce je popsat funkcionalitu pro zacházení s konfiguračními soubory sítě, viz kapitola 4 a popsat vytvořenou webovou aplikaci, viz kapitola 8.

## 2 Omezení původní verze

Toto je popis vlastností z původní verze modulu, které vedly k přepracování modulu.

### 2.1 Nemožnost diferenciací síťových prvků

Tuto vlastnost způsobovalo jednotné nastavení, do jakých konfigurací sítě se mají data ukládat. V konfiguraci modulu PC account, se nastavilo, zda se mají data ukládat do DHCP anebo DNS. Existovala, pouze možnost volby zda přidat záznam do NT, při manipulaci s konkrétním záznamem.

#### Příklad 2.1

Mějme uložit konfigurace účtu pro přepínač a klasický počítač. Pro přepínač chceme mít záznam pouze v DNS. Pro počítač chceme mít záznam v DHCP, DNS i NT doméně. ■

V tomto případě se musíme rozhodnout, zda chceme postrádat záznamy v konfiguraci DHCP a NT doméně pro počítač nebo aby přebýval záznam v DHCP pro přepínač. Nebo třetí možnost nevyužít modulu PC account a nakonfigurovat vše ručně, což bylo častým řešením.

### 2.2 Nepodpora více podsítí

Další omezující vlastností byla nepodpora více podsítí. Při konfiguraci modulu PC account se zvolilo číslo jedné z podsítí, tzv. UIDX, podle kterého přistupoval modul k podsítím v DHCP a DNS.

#### Příklad 2.2

Mějme server, který spravuje tři podsítě. Nazvěme je '3\_patro', '5\_patro' a 'archiv'. Chceme, aby administrátor mohl zacházet se všemi počítači v síti. Jaké máme možnosti? ■

Toto by mohla řešit instalace modulu do Webminu ve třech verzích, každá pro jinou podsít'. Jednoduché řešení, ale kdybychom měli spravovat síť s větším počtem podsítí, zřejmě by se toto jevilo jako dosti nepřehledné.

### 2.3 Dlouhá odezva při zobrazení dostupnosti síťových prvků pomocí aplikace Ping

Další, ne tolik omezující vlastností, byla dlouhá odezva při testování dostupnosti síťových prvků. Konkrétně nekomfortně dlouhé čekání při testování větší podsítě. Algoritmus tohoto testování probíhal sériově v témže procesu ve hlavním vlákne. Navíc celá aplikace prováděla dva druhy testování, pomocí klasické utility ping Ping, viz kap.3.9 a Arping, viz kap.3.10.

#### Příklad 2.3

Testujme dostupnost sta počítačů v jedné podsíti. Asi třetina je vypnuta a druhá třetina používá Windows XP SP2 nebo novější. Jak probíhá testovací proces? ■

Dojde k pozastavení generování HTML kódu, kde se v cyklech se zjišťuje dostupnost počítače, nejprve pomocí utility Ping a později pomocí Arping . V případě použití utility Ping, kde počítač není dostupný, se po vyslání prvního ICMP paketu a vypršení jeho platnosti, vyšlou se ještě další tři a čeká se na vypršení jejich platnosti. K dalšímu zdržení dochází při testování počítačů pomocí utility Arping u počítačů u kterých už víme, že jsou dostupné.

## 3 Použité technologie

V této kapitole se pokusím shrnout a nastínit technologie, které byly využity při implementaci projektu. Přednosti jednotlivých technologií, kvůli kterým byla technologie vybrána. Nejedná se o všeobecný popis technologie, pouze jakési shrnutí, aby se čtenář mohl přímo představit problematiku a nemusel se ztrácet v manuálech a technických specifikacích.

### 3.1 Webmin

Webmin je webové rozhraní pro administraci systému Unix. Použitím webového prohlížeče můžeme spravovat uživatelské účty, Apache, DHCP, DNS, sdílení souboru aj. S pomocí Webminu odpadá potřeba ruční editace konfiguračních souborů typu */etc/passwd*, místo ní můžeme konfigurovat server pomocí webového rozhraní vzdáleně a interaktivně.

Po nainstalování služby Webmin na serveru, můžeme využívat kromě standardních modulů ještě moduly třetích stran nebo si napsat své vlastní. Webmin je napsaný na technologii CGI a vnitřně používá skriptovací jazyk Perl.

Velké plus Webminu je viditelné, když chceme konfigurovat server, ke kterému nemáme fyzický přístup, kde v rámci topologie sítě je server umístěn za Proxy serverem, který zahazuje příchozí spojení typu SSH. Přes webové rozhraní spouštíme funkcionalitu, která se vykoná na daném serveru a přehledně se zobrazí výsledek dané akce.

### 3.2 SQLite

SQLite je relační databázový systém obsažený v relativně malé knihovně napsané v jazyku C. Je vyvíjen D. Richardem Hippem a šířen pod licencí public domain.

Na rozdíl od databází založených na principu klient-server, kde je databázový server spuštěn jako samostatný proces, je SQLite pouze malá knihovna, která se linkuje k aplikaci a pomocí jednoduchého rozhraní ji lze začít využívat. Každá databáze je uložena v samostatném souboru *.dbm* (Database Manager), kde se data ukládají za použití jednoduchého primárního klíče do stejně velkých bloků a používá hashovací techniky pro rychlý přístup k datům při vyhledávání podle klíče.

V SQLite je implementován téměř celý standard *SQL-92*. Databázi SQLite lze použít například v programovacích jazycích C, C++, Delphi, PHP, Java, Python, Perl, Tcl aj. SQLite nemá být náhradou za větší produkty typu PostgreSQL, Oracle, aj., ale dobře se osvědčil jako náhrada za funkci *fopen()*.

### 3.3 Perl

Perl je interpretovaný programovací jazyk vytvořený Larry Wallem v roce 1987. S rozvojem internetu se Perl stal velmi populárním nástrojem pro tvorbu CGI skriptů.

Perl zahájil svou éru jako skriptovací jazyk, náhrada jazyka AWK a interpretu sh. Největšího rozšíření dosáhl ve verzi 4 z roku 1991. Verze 5 přinesla četná vylepšení,

především výkonné datové struktury a možnost objektového programování. Slovo Perl najdete v Oxford English Dictionary.

Jazyk umožňuje psát krátké programy jednoduše a rychle a přitom nebrání v psaní těch složitých. Vždy je více způsobů jak dosáhnout cíle a jeden ze způsobů je přitom obvykle velmi stručný, čímž Perl získal nezaslouženou pověst jazyka, ve kterém se tvoří nesrozumitelný a neudržovatelný kód. Tato kritika ale není oprávněná, Perl je vhodný k řešení malých i velkých problémů. Schopnosti a nástroje, které se používají u velkých projektů, lze použít i v krátkých skriptech.

### 3.4 CPAN

CPAN, neboli Comprehensive Perl Archive Network je archív více než 14 800 modulů napsaných v Perlu nebo pro Perl, všechny přehledně zdokumentované. Nachází se na stránkách [www.cpan.org](http://www.cpan.org) a je zrcadlen ve více než 250 lokacích. Většina modulů CPANu je svobodný software.

### 3.5 Šablonovací systém HTML::TEMPLATE::Compiled

HTML::TEMPLATE::Compiled je jeden z více modulů z archívu CPAN sloužících pro práci s HTML šablonami. Motivace použití - oddělení grafiky od aplikační logiky. Do souboru se šablonou jsou vloženy HTML značky do kterých se v průběhu vykonávání programu přiřazují hodnoty. Hodnoty a šablona se při zobrazení linkují ve výsledné HTML.

Tento modul podporuje základní smyčky, procházení polí, podmínky, aj. HTML kód tedy můžeme generovat dynamicky. Oddělenou šablonu pak zvládne editovat i grafik bez znalosti programovacího jazyka. Více informací v [5].

### 3.6 DHCP

DHCP je aplikační protokol z rodiny TCP/IP. Používá se pro automatické přidělování IP adres počítačům v počítačových sítích, čímž zjednodušuje jejich správu.

DHCP je široce konfigurovatelné, nás zajímá konfigurační soubor v *dhcpd.conf*, kde jsou do skupin podle podsítí, definované statické přidělené IP adresy počítačů, jejich hostname a MAC adresa.

Další, co je z pohledu modulu PC account důležité jsou informace ze souboru *dhcpd.leases* ve kterém si implementace DHCP ukládá informace o dynamicky přidělených adresách, opět IP, MAC adresu, jméno síťového prvku, dobu platnosti zapůjčení adresy, atd.

### 3.7 DNS

DNS je hierarchický systém doménových jmen, který je realizován servery DNS a protokolem stejného jména, kterým si servery vyměňují informace. Jeho hlavním úkolem a příčinou vzniku jsou vzájemné převody doménových jmen a IP adres uzlů sítě. Později

přibral další funkce (např. pro elektronickou poštu či IP telefonii) a slouží dnes jako distribuovaná databáze síťových informací.

Jména domén umožňují lepší orientaci lidem, adresy pro stroje jsou však vyjádřeny pomocí adres 32bitových (IPv4) A záznam nebo 128bitových (IPv6) - AAAA záznam. Systém DNS umožňuje efektivně udržovat decentralizované databáze doménových jmen a jejich překlad na IP adresy. Stejně tak zajišťuje zpětný překlad IP adresy na doménové jméno - PTR záznam.

Z pohledu PC account modulu se do DNS ukládají informace o jménu síťového prvku a IP adresa do dvou druhů záznamu: A záznam a PTR záznam. Tyto záznamy se ukládají strukturovaně podle podsítě, do které spadají.

V konfiguračním souboru *named.conf* se spravují podsítě. Existují vždy dva záznamy. Jeden pro reverzní překlad a druhý pro doménové jméno. Každý obsahuje záznam *file* se jménem souboru ve složce *\etc\namedb*, ve kterých najdeme jednotlivé informace o jménech prvků a IP adresách.

---

```
zone "30.20.10.in-addr.arpa" {
    type master;
    file "10.20.30";
    allow-update { localhost; };
};

zone "tonybox.cz" {
    type master;
    file "tonybox.cz";
    allow-update { localhost; };
};
```

---

Výpis 1: Ukázka konfiguračního souboru DNS z namedb

### 3.8 Záznam v NT doméně

Záznam v NT doméně je typ záznamu do modulu Samba, pro sdílení a komunikaci unixových systémů se systémy Microsoft. Informace pro NT domény se ukládají do souboru *\etc\passwd*.

Windows Server domain je logická skupina počítačů fungující na operačních systémech Microsoft Windows, které sdílí centrální adresářovou strukturu. Tato centrální databáze (známá jako Active Directory od verze Windows 2000, nebo jako NT Directory Services na Windows NT Server) obsahuje uživatelské konta a informace zdrojů v této doméně. Každá osoba která využívá počítač v doméně obdrží své vlastní unikátní uživatelské jméno. Tyto účty pak mohou přistupovat k doménovým zdrojům.

### 3.9 Ping

Program Ping umožňuje prověřit funkčnost spojení mezi dvěma síťovými rozhraními (počítače, síťová zařízení) v počítačové síti, která používá rodinu protokolů TCP/IP. Ping při své činnosti periodicky odesílá IP datagramy a očekává odezvu protistrany. Při úspěšném obdržení odpovědi vypíše délku zpoždění a na závěr statistický souhrn.

Parametrem programu ping je doménové jméno nebo IP adresa síťového rozhraní, jehož dostupnost chceme prověřit. Je-li uvedeno doménové jméno, je nejprve přeloženo pomocí DNS na IP adresu. Program využívá zprávy "Echo Request" (výzva) a "Echo Reply" (odpověď) protokolu ICMP. Výzvy jsou odesílány na cílovou IP adresu a ve stanoveném limitu se očekává odpověď (typicky 3 sekundy). Jednotlivé výzvy obsahují čísla (`icmp_seq`), podle kterých je možné identifikovat jednotlivé odpovědi nebo jejich ztrátu. Program průběžně vypisuje, které odpovědi již došly a s jakým zpožděním (latencí).

### 3.10 Arping

Program Arping je funkcionalitou podobný programu Ping, ale místo protokolu ICMP pracuje s ARP. Protože využívá ARP protokol, je použitelný pouze pro lokální síť. V některých případech můžeme získat odezvu ne od prvku, který testujeme ale od prvku který si mac rezervoval v proxy ARP (router).

Nicméně Arping je užitečný v zjišťování dostupnosti počítačů s Windows XP SP2 a pozdějších, kde Windows Firewall zahazuje ICMP pakety.



## 4 Zacházení s konfiguračními soubory

Zacházení s konfiguračními soubory DHCP, DNS a NT domény, jsou základem celého projektu. Tuto funkčnost již podporovala minulá verze. Mým rozšířením této funkčnosti bylo obalení volaných funkcí do nových funkcí, které jsou uloženy v knihovně tohoto modulu *pcaccount-lib.pl*. Tím se dosáhlo větší modulárnosti a intuitivnějšího zacházení s konfiguracemi.

### 4.1 Zacházení s DHCP konfiguračními soubory

Pro zacházení s konfiguračními soubory DHCP slouží tyto základní tři funkce. Ostatní funkce jsou popsány v příloze.

Při manipulaci s prvku DHCP, jsou pro nás klíčové parametry jména prvku, IP adresa a MAC adresa.

- `&dhcpNewNode( $node, $ip, $mac, $hw_type, $uidx );`  
Přidání nového prvku, do konfigurace DHCP. Parametry jsou jméno prvku, IP adresa, MAC adresa, hardware typ, a číslo podsítě.
- `&dhcpEditNode( $old_node, $node, $ip, $mac, $hw_type, $uidx, $old_uidx );`  
Editace stávajícího prvku, s parametry stávající jméno prvku, nové jméno prvku, IP adresa, MAC adresa, hardware typ, číslo podsítě a stávající číslo podsítě.
- `&dhcpRemoveNode( $node );`  
Odstranění existujícího prvku s parametrem jména prvku na odstranění.

#### 4.1.1 Zobrazení prvků z leases

Jednou z funkcionalit, tohoto modulu je zobrazování počítačů, které mají zapůjčenou adresu z adresního poolu modulu DHCP. Toto se děje parsováním souboru se zapůjčenými adresami, standartně `\var\lib\dhcpd\dhcpd.leases`. Do tohoto souboru jsou ukládána data tímto způsobem:

---

```
lease 10.20.30.40 {
    starts 2 2008/11/13 05:34:15;
    ends 2 2008/12/20 17:34:15;
    tstp 2 2008/11/20 23:34:15;
    binding state active;
    next binding state free;
    hardware ethernet 00:11:22:33:44:55;
    uid "\001\000\025\305\312\342\252";
    client -hostname "pcTonik";
}
lease 10.20.30.41 {
    starts 2 2008/11/13 06:07:26;
    ends 2 2008/12/20 18:07:26;
    tstp 2 2008/11/20 18:07:26;
```

---

```

binding state active;
next binding state free;
hardware ethernet 00:11:22:33:44:56;
uid "\001\000\026>D\013\000";
client -hostname "pcTomik";
}

```

---

#### Výpis 2: Ukázka souboru zapůjčených adres DHCP

Data jsou následně získána pomocí funkcí *&tokenize\_file* a *&parse\_struct* z knihovny *dhcpcd-lib.pl*. Z těchto hodnot získáváme jméno prvku, IP a MAC adresu, které přidáváme do seznamu síťových prvků na titulní stránce modulu PC Account. Tyto prvky jsou barevně odlišeny a mohou být po změně IP adresy importovány mezi ostatní záznamy se staticky přidělenou IP adresou.

## 4.2 Zacházení s DNS konfiguračními soubory

Pro zacházení s konfigurací DNS máme opět tři základní funkce pro přidání nového prvku, editaci prvku a smazání. Pro jejich využití, ale musíme získat další informace, z funkce *&dnsGetDom( \$uidx )*; UIDX je zde číslo podsítě.

Při manipulaci s prvkem DNS, jsou pro nás klíčové parametry jména prvku a IP adresa.

---

```

$uidx=5;
my $dom = &dnsGetDom( $uidx );
&dnsNewNode( $dom, 'pcTonik', '10.20.30.40' );

```

---

#### Výpis 3: Přidání nového prvku do DNS

---

```

$uidx=5;
my $dom = &dnsGetDom( $uidx );
&dnsEditNode( $dom, 'pcTomik', 'pcTonik', '10.20.30.40' );

```

---

#### Výpis 4: Editace prvku v DNS

---

```

$uidx=5;
my $dom = &dnsGetDom( $uidx );
&dnsRemoveNode( $dom, 'pcTonik' );

```

---

#### Výpis 5: Výmaz prvku z DNS

---

### 4.3 Zacházení s `passwd`

Zacházení se souborem *passwd*, do kterého se ukládají informace pro použití v NT doméně, se děje podobnými funkcemi jako u DHCP či DNS. Funkce pro přidání nového, editaci existujícího a smazání záznamu.

---

```
# přidá nový
&ntNewNode('pcTomik');

# editace záznamu
&ntEditNode( 'pcTomik', 'pcTonik' );

# smazání záznamu
&ntNewNode('pcTonik');
```

---

Výpis 6: Ukázka zacházení se záznamy v NT doméně

---

```
pcTonik\$:x:10001:500:::/bin/false
pcTomik\$:x:10001:500:::/bin/false
```

---

Výpis 7: Ukázka záznamů v *passwd*

Bližší informace o souboru `passwd`, viz [3, 2];

## 5 Zavedení centrální relační databáze

Zavedením centrální relační databáze získáváme výhodu uložení nastavení konfiguračních souborů sítě v jediné databázi. Z této databáze pak můžeme například:

- synchronizovat konfigurační soubory s databází
- rozšířit uchovávané informace o jednotlivých síťových prvcích, které by neměly být uloženy v konfiguračních souborech
- vytvářet pravidelné zálohy stavu konfigurací

K tomuto účelu se nejlépe hodí databáze SQLite, viz kap.3.2. Celá databáze je pak uložena na disku serveru jako jeden soubor. Databázi může využívat jak tento modul při úpravách konfigurace, tak i ostatní moduly nebo procesy, které z databáze mohou získat užitečné informace. Celé databázové schéma je zobrazeno ve formě ER diagramu, jak si jej představuje Microsoft Visio na obr.1.

### 5.1 Uložení informací o síťových prvcích /nodes/

Základem této databáze jsou informace o síťovém prvku, node. O něm evidujeme jeho název, IP adresu, MAC adresu a krátký dodatečný popis. Tyto informace ukládáme do tabulky *Nodes*. Prvek je vždy členem jednoho profilu. Význam profilu, viz kap.5.2.

### 5.2 Zavedení profilů

Zavedení použití profilů, je vývojový krok tohoto modulu, kterým se řeší problematika vlastností předchozí verze, která zabráňovala direferenciaci typu zařízení, viz. kap.2.1.

Nadefinováním vlastních profilů a jejich přiřazení k jednotlivým prvkům můžeme odlišit, do kterých konfigurací se mají data o prvcích ukládat. Zde máme na výběr využití DHCP, DNS i NT domény.

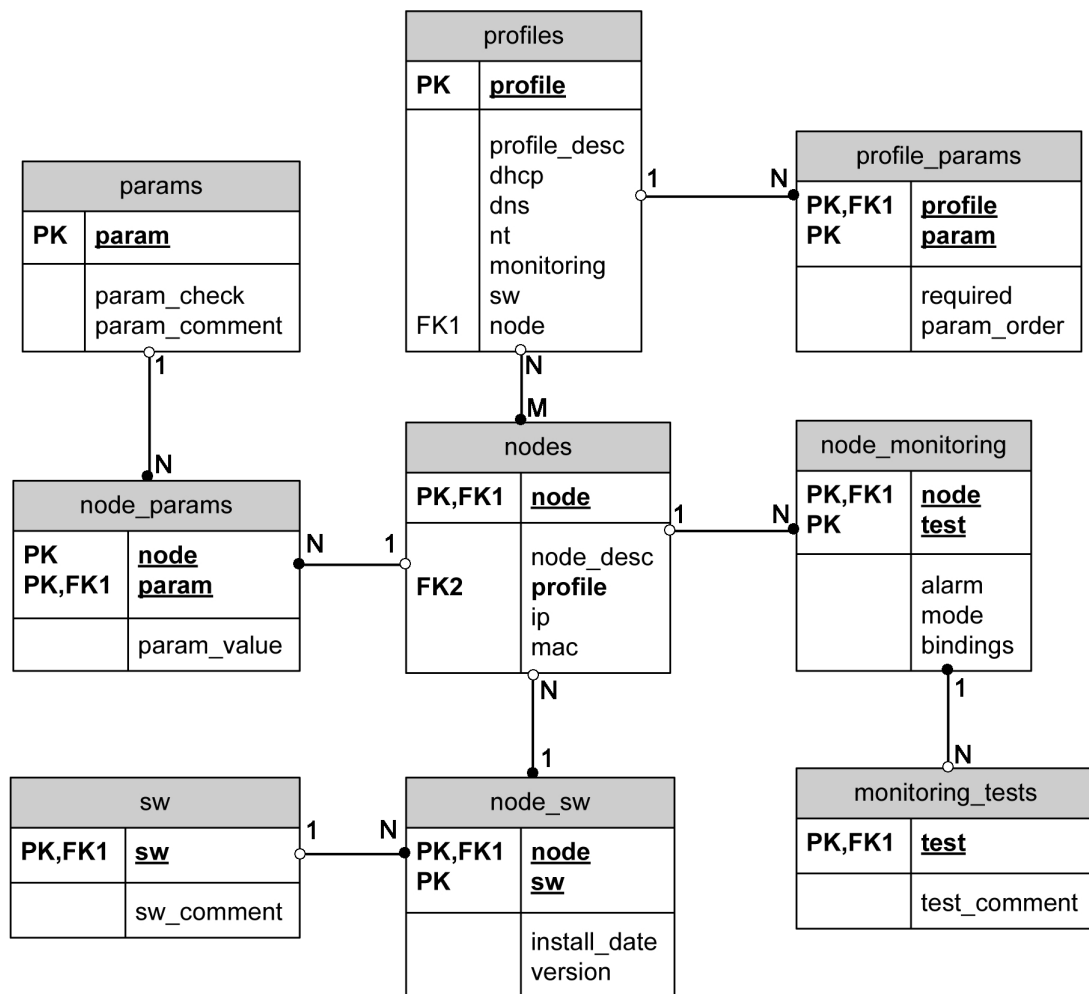
Navíc je v modulu přidána funkčnost pro evidování výsledků monitoring testů a software nainstalovaného na evidovaném prvku, viz kap.5.3.2. Jejich povolením v profilu tedy povolíme funkcionalitu využívat těchto dodatečných parametrů pro prvky tohoto profilu.

### 5.3 Doplnující informace o prvcích

K uložení celého spektra všech druhů dodatečných informací, které by si uživatel modulu chtěl o prvcích uchovat, používáme tři druhy dodatečných parametrů.

#### 5.3.1 Profilové parametry

Profilové parametry jsou dotatečné parametry, které dávají uživateli možnost, pro uložení jakéhokoli množiny parametry pro jednotlivé prvky.



Obrázek 1: Databázové schéma

- Nejprve se nadefinují jednotlivé parametry do tabulky *params*. Zde se uloží název parametru - *param*, podle čeho se má parametr validovat - *param\_check*, a krátký doplňující komentář k parametru - *param\_comment*.
- Dále se do tabulky *profile\_params*, uloží sledování profilových parametrů pro jednotlivé profily. Zde tedy ukládáme, název profilu - *profile* a název parametru *param*. Dále zda je parametr povinný - *required* a jeho pořadí při zobrazování formuláře - *param\_order*.
- Podledním krokem je uložení jednotlivých hodnot pro každý prvek z množiny jednoho profilu. To se děje v tabulce *node\_params*. Zde ukládáme hodnotu názvu síťového prvku - *node*, název parametru - *param* a zvalidovanou hodnotu parametru - *param\_value*.

Použitím profilových parametrů, získáváme výhodu maximální flexibility evidovaných parametrů, za cenu větší režie při práci s nimi. Uložení parametrů je vyřešeno relačně a tudíž nemusíme upravovat schéma databáze, při každém požadavku na nový parametr.

### 5.3.2 Monitoring testy a Software parametry

Využitím těchto druhů parametrů o prvcích, získáváme možnost uložení výsledků monitoring testů a nainstalovaného software. S těmito parametry se počítalo při návrhu databáze a jejich pravý účel vyplývá s ostatních modulů, které se instalují na servery zákazníků.

Pro oba typy parametru jsou obdobné. Tabulka *Monitoring*:

- V tabulce *monitoring\_tests* máme číselník s monitoring testy. Zde ukládáme název testu - *test* a jeho krátký popis - *test\_comment*.
- V tabulce *node\_monitoring* spojujeme informace o testech a uzlech. Přidáváme další informace, zda na test existují další vazby *bindings*, styl testu *mode* a zda má být na test upozorováno *alarm*. O tyto hodnoty se starají jiné moduly.

Tabulka *Software*:

- V tabulce *sw* máme číselník s nasazovaným software. Zde ukládáme název software a jeho krátký popis - *sw\_comment*.
- V tabulce *node\_sw* spojujeme informace o software a uzlech. Přidáváme další informace, datum instalace *install\_date* a verzi software *version*.

## 5.4 Zavedení skupin

Motivace pro zavedení skupin je taková, aby si uživatel-administrátor mohl rozdělit a třídit prvky v seznamu na úvodní stránce modulu podle vlastního uvážení, např. podle lokalit: '3\_patro', '5\_patro', 'archív'. Tyto skupiny jsou do databáze ukládány jako profilové parametry, viz kap.5.3.1, je nutno je tedy po instalaci modulu nastavit.

## 5.5 Detekce chybějících záznamů

Detekce chybějících záznamů má za úkol odhalit nekonzistence mezi relační databází a jednotlivými konfiguračními soubory. Důvodem může být například nešikovný zásah administrátora do jednoho z konfiguračních souborů. Tím by se sít začít chovat jinak, než by jsme podle našeho modulu čekali.

Detekce probíhá při zobrazování úvodní stránky, kdy se pomocí funkcí *&dhcpGetAllNodes()*, *&dnsGetAllNodes()* a *&ntGetAllNodes()* porovnávají vrácené hodnoty s hodnotami v relační databázi. Porovnáním jednotlivých uzlů můžeme získat tyto výsledky:

- prvek nemá záznam v DHCP/DNS/NT doméně
- prvek má záznam v DHCP/DNS/NT doméně, ale jeden z údajů se neshoduje
- prvek má záznam v DHCP/DNS/NT doméně, který nefiguruje v databázi
- prvek má záznamy tam kde je potřeba

První tři stavy jsou nežádoucí a je potřeba opravit je. Toto má na starosti následující kapitola.

## 5.6 Synchronizace konfiguračních souborů s databází

K dosažení konzistence mezi databází a konfiguračními soubory, jsou naimplementovány tři algoritmy.

### 5.6.1 Nativní uložení

Při ukládání, importování či editaci záznamu dochází k uložení, smazání či editaci záznamu v konfiguračních souborech DHCP, DNS a NT domény, podle zvoleného profilu, čímž by se dalo předpokládat, že budou data synchronní i nadále. Tento předpoklad je ale špatný, viz kap.5.5.

### 5.6.2 Celková synchronizace

Celková synchronizace je naimplementována podle původních požadavků, aby se mohla spouštět ihned po detekci chybějících případně i přebývajících záznamů v konfiguracích. Při původním zadání se mělo za to, že relační databáze je primárním úložištěm dat a tudíž jakékoliv změny provedené přímo v konfiguračních souborech budou při synchronizaci smazány. Toto chování se časem ukázalo jako nepřijatelné a tudíž od něj bylo upuštěno.

### 5.6.3 Současné řešení

Současným řešením jak dostat databázi do synchronního stavu s konfiguračními soubory je takové, že při zjištění chybějících, chybných či přebývajících záznamů v konfiguracích se zobrazí informace o tomto chybném stavu na úvodní straně s výzvou aby uživatel,

importoval chybějící počítače, bohužel stylem importu po jednom záznamu a upravil nekonzistentní záznamy. Toto řešení zřejmě není moc efektivní, ale pro zadavatele postačující.



## 6 Podpora podsítí

Podpora podsítí je jedna z vlastností, které byly do tohoto modulu přidány po původní definici zadání. Je to reakce na popisované nežádoucí chování, viz kap.2.2. Pokud chceme spravovat více, podsítí z jednoho modulu, je potřeba mírně poupravit stávající funkcionalitu, tím že při vkládání, editaci či importu síťového prvku do modulu, přidáme na výběr do které podsítě chceme prvek uložit.

### 6.1 Informace o podsítích

Funkcí *&getSubnets()*, získáme všechny informace o podsítích, jejich číslo podsítě UIDX, masku, rozsah IP adres, jméno aj.

### 6.2 Ukládání prvku do určité podsítě

Co se týče konfiguračních souborů, o podsítě nám jde jen v konfiguracích DHCP a DNS. Při ukládání a celkově při práci s prvkem se tedy nejedná podle čísla podsítě, nastaveného v konfiguraci modulu PC Account, ale tímto zvoleným číslem podsítě UIDX.

### 6.3 Přesun prvku do jiné podsítě

Uložení a smazání síťového prvku z konfigurace DHCP či DNS se jeví jako jednoduchá operace. K mírným komplikacím docházelo při návrhu přesouvání mezi podsítěmi. Řešení tohoto problému ale nebylo moc náročné, stačí smazat prvek v staré podsíti a vložit nový prvek do nové podsítě. Tato funkcionalita je samozřejmě zahrnuta ve funkcích *&dhcpEditNode* a *&dnsEditNode*.

## 7 Paralelní zjišťování dostupnosti síťových prvků

Reakcí na nevyhovující vlastnost minulého modulu dlouhého čekání na odezvu dostupnosti síťových prvků, viz kap.2.3 je paralelní implementace tohoto procesu. Požadavem bylo zbytečně nepřetěžovat server velkým počtem vláken a tudíž bylo zvoleno 10 samostatně testujících vláken.

### 7.1 Samostatná aplikace Pingu mimo proces Webminu

Pro proces paralelního testování dostupnosti síťových prvků tedy byl navržen skript v jazyce Perl, který se spouští mimo proces Webmin. Skript s názvem *arping.pl* se spouští jako samostatný proces z procesu Webmin, kterému po dokončení předá své výsledky. Skript *arping.pl* funguje tak, že v procesu spustí 10 vlákek, které postupně testují dostupnost prvku pomocí utility Arping. Po dokončení testu dostupnosti všech prvků skript vypisuje na standartní vstup data ve specifické podobě. Ty převezme Webmin, regulárními výrazy si vybere výsledky a ty se později zobrazují.

### 7.2 Důvody využití příkazu Arping

Důvodem využití utility Arping je právě využití protokolu ARP. S pomocí tohoto protokolu dokážeme ověřit stav počítače fungujícím na Windows XP SP2 a novějších, u kterých by kvůli standartnímu nastavení Windows Firewall klasický Ping s ICMP protokolem nefungoval.

## 8 Webová aplikace

Po instalaci potřebného software na server, spuštění služby Webmin a instalaci modulu, můžeme spustit webovou aplikaci. Ta běží na *http://adresa\_serveru:10000/lbox\_pcaccount/*. Zde po zalogování k aplikaci Webmin dostáváme kompletní přístup k modulu.

### 8.1 Seznam prvků

Na úvodní stránce *index.cgi*, nalezneme seznam síťových prvků, které modul PC account obsluhuje. Na stránce se nabízí následující možnosti:

- Přidat stanici  
Zobrazí formulář pro přidání nového síťového prvku.
- Zobrazit včetně pingu na PC  
Znovu vykreslí stránku *index.cgi* se zobrazením dostupnosti jednotlivých síťových prvků.
- Import  
Znovu vykreslí stránku *index.cgi* se zobrazením počítačů, které nejsou uloženy v relační databázi, ale jsou v konfiguračních souborech.
- Správa profilů  
Přejde na stránku *profiles.cgi*, kde lze nastavit všechny detaily ohledně profilů.
- Tlačítko smazat vybraná PC  
Smaže informace z relační databáze i konfiguračních souborů o síťových prvcích vybraných z tabulky síťových prvků.
- Třídění podle jednoho atributu v odkazu hlavičky tabulky  
Seřadí tabulku síťových prvků podle požadovaného atributu.
- Víceúrovňové třídění pod tabulkou  
Seřadí tabulku podle prvního vybraného atributu, v případě duplicit v tomto sloupci, se seřadí podle druhého atributu.

### 8.1.1 Třídění podle dvou atributů

Tuto funkcionalitu hodnotím jako docela zajímavou, a proto vkládám kód vícenásobného třídění.

---

```

foreach $node (
  sort {
    # první uroveň třídění
    &my_cmp(
      ( $in{'sort1_num'} == -1 ? $a : $nodes->{$a}->{$in{'sort1'}} ),
      ( $in{'sort1_num'} == -1 ? $b : $nodes->{$b}->{$in{'sort1'}} ),
      $in{'sort1'}
    )

    || # a zde je ta zajímavost, obvyčejné nebo mezi dvěma funkcemi třídění

    # druhá uroveň třídění
    &my_cmp(
      ( $in{'sort1_num'} == -1 ? $a : $nodes->{$a}->{$in{'sort2'}} ),
      ( $in{'sort1_num'} == -1 ? $b : $nodes->{$b}->{$in{'sort2'}} ),
      $in{'sort2'}
    )
  }
  keys %$nodes
){
  # přidej informace o požadovaném prvku do pole pro zobrazení v šabloně
}

```

---

Výpis 8: Ukázka třídění podle dvou atributů

## 8.2 Manipulace se záznamy

Na obr.2 je zobrazeno rozhraní pro manipulaci se záznamy, přes které se manipuluje se záznamy. Toto rozhraní se generuje dynamicky podle vybraného profilu. Slouží pro vkládání nových, editaci stávajících, import a případně i výmaz síťových prvků.

### 8.2.1 Monitoring testy

Parametry pro monitoring testy se vkládají do tabulky *Monitoring testy*. Z rozevíracího seznamu se vybere, o kterém testu jsou vedeny informace, doplní se atributy alarm, režim a vazby, které mají návaznosti na jiné moduly. GUI je konfigurováno pro vkládání tří záznamů o monitoring testech, na každé zobrazení stránky. Tj. při vkládání více např. čtyř monitoring testů musíme, editovat záznam dvakrát.

### 8.2.2 Parametry týkající se software

Parametry o software se vkládají do tabulky *Software*. Z rozevíracího seznamu se vybere software, který je nainstalován, doplní se datum instalace a verze. GUI je konfigurováno pro vkládání tří záznamů o software, na každé zobrazení stránky. Tj. při vkládání více např. čtyř kusů software musíme, editovat záznam dvakrát.

[Index](#) » [PC stanice](#) » PC stanice:

## Detaily stanice

Parametry počítače	
Název PC	pcTonik
Popis počítače	Tonikův počítač
Profil	dhcp_only <input type="button" value="Načti"/>
Podsít'	linuxbox.cz
IP adresa	10.20.30.40
Hardware adresa	11-22-33-44-55-66

\* nutno vyplnit

Software			
Software	Datum instalace	Verze	Smazat
AdobeReaderR	2.2.2222	5	<input checked="" type="checkbox"/>
AdobeReaderR	6.5.2009	9	-
MS_Office	6.5.2009	2003	-
-			-

Monitoring testy			
Test	Alarm	Režim	Vazby
ping	monthly		ano
snmp	monthly		ne
-			ne

[Zpět na seznam](#)

Obrázek 2: Ukázka GUI pro manipulaci se záznamy

[Index](#) » [PC stanice](#) » Správa profilů:

## Editace profilu

Parametry:	
Název PC	<input type="text" value="pocka"/>
Popis profilu	<input type="text"/>
Zaznam v dhcp	<input type="radio"/> ne <input checked="" type="radio"/> ano
Zaznam v dns	<input type="radio"/> ne <input checked="" type="radio"/> ano
Zaznam v NT domene	<input type="radio"/> ne <input checked="" type="radio"/> ano
Monitoring	<input type="radio"/> ne <input checked="" type="radio"/> ano
Software	<input type="radio"/> ne <input checked="" type="radio"/> ano

Doplňkové atributy:				
	Atribut	Popis atributu	Povinný	Presun
<input checked="" type="checkbox"/>	skupina		<input type="radio"/> ne <input checked="" type="radio"/> ano	↑↓
<input checked="" type="checkbox"/>	serial_num	seriove cislo	<input checked="" type="radio"/> ne <input type="radio"/> ano	↑↓
<input type="checkbox"/>	uidx	číslo podsítě	<input type="radio"/> ne <input type="radio"/> ano	
<input type="checkbox"/>	popiska		<input type="radio"/> ne <input type="radio"/> ano	

[↩ Zpět na profily](#)

Obrázek 3: Ukázka GUI pro manipulaci s profily

### 8.2.3 Profilové parametry

Profilové parametry jsou zobrazeny v tabulce *Parametry počítače*, pod atributem *Hardware adresa*. Jedná se o parametry, popsané v kap.5.3.1. Jsou zobrazovány dynamicky podle vybraného profilu.

## 8.3 Správa profilů

Z *index.cgi* vede hypertextový odkaz *Správa profilů* na stránku *profiles.cgi*. Zde se nabízí webové rozhraní pro:

- vytváření, editaci a mazání profilů
- vytváření, editaci a mazání profilových parametrů
- vytváření, editaci a mazání monitoring testů
- vytváření, editaci a mazání používaného software
- přiřazování profilových parametrů, parametrů pro monitoring anebo parameterů o software

## 9 Závěr

V této práci jsem se snažil navrhnout a implementovat modul do aplikace Webmin, tj. vytvořit webovou aplikaci s funkcemi pro konfiguraci modulů DHCP, DNS a NT domény. Modul se základní funkcionalitou, který jsem převzal, jsem se snažil co nejvíce zmodularizovat, a upravit pro co nejjednodušší zacházení s konfiguračními soubory. S použitím šablonovacího systému se podařilo se oddělit aplikační logiku od prezentace, čímž se podařilo zpřehlednit zdrojové kódy pro další úpravy.

### 9.1 Vývoj do budoucna

Předpokládám, že vývoj tohoto modulu není u konce. A dle mého názoru, by se hodilo přepracovat návrh synchronizace s použitím výběru, jak nakládat s nesynchronními záznamy.



## 10 Reference

- [1] *Webmin*, <http://www.webmin.com/>
- [2] *Wikipedia - free encyclopedia*, <http://wikipedia.org/>
- [3] *Unix manual pages*
- [4] *Comprehensive Perl Archive Network*, <http://www.cpan.org/>
- [5] CPAN *HTML::Template::Compiled* module documentation,  
<http://search.cpan.org/perldoc?HTML::Template::Compiled>
- [6] *SQLite*, <http://www.sqlite.org/>

## A Ostatní funkce pro zacházení s konfiguračními soubory

Toto je seznam funkcí využívaných v hlavních funkcích pro přidání, editaci a výmaz prvku.

- `&dhcpGetAllNodes();`  
Vrací pole polí všech prvků s informacemi o jméně prvku, IP adrese, MAC adrese, a číslu podsítě UIDX.
- `&dhcpCheckDuplicityIp( $old_node, $ip );`  
Kontrola, zda u již existujících prvků v DHCP neexistuje prvek se stejnou IP adresou.
- `&dhcpCheckDuplicityMac( $node );`  
Kontrola, zda u již existujících prvků v DHCP neexistuje prvek se stejnou MAC adresou.
- `&dhcpCheckDynamicRange( $ip );`  
Kontrola, zda IP adresa nespadá do rozsahu dynamicky přidělovaných adres.
- `&dhcpCheckIpSubnet( $ip );`  
Kontrola, zda IP adresa spadá do rozsahu podsítě a poslední byte není roven 0 nebo 255.
- `&dnsGetAllNodes( $uidx );`  
Vrací pole polí všech prvků s informacemi o jméně prvku a IP adrese.
- `&dnsGetPtrFileName( $uidx );`  
Získá jméno souboru pro DNS PTR záznamy.
- `&dnsGetRecords( $dom );`  
Vrací pole asociativních polí všech záznamů v DNS a jméno souboru pro A záznamy.
- `&ntGetAllNodes();`  
Vrací polí všech prvků.